CONNECTMYAPPS AS

Penetration Test Summary 2025

Abstract

Summary of penetration test performed during 2025.

Contents

Change Log 2

Contacts 2

Introduction 3

What this Summary Contains 3

Who Performed the Test 3

What Was Tested 3

When the Test Was Conducted 5

Vulnerability Classification 5

Summary of Test Results 7

Change Log

Changed By	Date	Version	Note
Igor Kostian	2025-12-12	1.0	Approved for distribution.

Contacts

Contact	Role	Email
Igor Kostian	CTO	Igor.kostian@connectmyapps.com

Introduction

At ConnectMyApps, security is our top priority. To protect our users and data, we engage a third-party cybersecurity expert to perform a comprehensive **Web Application Penetration Test** on an annual basis. The purpose of this test is to assess for potential vulnerabilities in our application, APIs, and infrastructure, enabling us to proactively enhance our systems and ensure secure, reliable solutions for our customers.

What this Summary Contains

This summary describes the details of the testing, including:

- Who performed the tested.
- What was tested, and when.
- Classification of any vulnerabilities found.
- Summary of the results.

For security purposes, this summary does not contain detailed test descriptions, details of ConnectMyApps infrastructure, or specific details of any vulnerabilities identified.

Who Performed the Test

The testing was conducted by a third-party cybersecurity expert, DigitalXRAID, located at:

Suite 9A Cavendish Court
South Parade
Doncaster
DN1 2DJ
United Kingdom

Web address: https://www.digitalxraid.com/.

What Was Tested

The penetration test conducted by DigitalXRAID provided a comprehensive evaluation of our web application, APIs, and external infrastructure. The assessment focused on identifying vulnerabilities from both authenticated and unauthenticated perspectives, ensuring thorough coverage across all layers of the system.

DigitalXRAID utilized a combination of industry-standard tools, techniques, and advanced methodologies refined through years of experience. The testing approach was closely aligned with

the **Open Web Application Security Project (OWASP) Top 10**, the **Open-Source Security Testing Methodology Manual (OSSTMM)**, and other recognized best practices in cybersecurity.

The **OSSTMM**, a globally respected framework for security testing, guided key elements of the assessment. It provided a structured methodology for evaluating the application's attack surface, including:

- **Rules of Engagement**: Ensuring clear communication, transparency, and ethical testing practices throughout the process.
- **Trust Analysis**: Examining how trust relationships within the system could be exploited by malicious actors.
- **Information Security**: Assessing the confidentiality, integrity, and availability of sensitive data.
- **Process Security**: Evaluating operational workflows to identify potential risks in application logic and processes.
- **Controls Verification**: Testing the effectiveness of technical and procedural controls in place to secure the system.

The **OWASP Top 10**, a cornerstone for web application security, outlines the most critical risks and informed key areas of focus during the assessment, including:

1. **Broken Access Control** – Ensuring proper restrictions on user permissions and access.
2. **Cryptographic Failures** – Reviewing encryption protocols to protect sensitive data.
3. **Injection Attacks** – Detecting vulnerabilities like SQL injection, cross-site scripting, and others.
4. **Insecure Design** – Evaluating system architecture for inherent security flaws.
5. **Security Misconfigurations** – Identifying improper or weak configurations.
6. **Vulnerable and Outdated Components** – Checking for unpatched software or dependencies.
7. **Identification and Authentication Failures** – Assessing login processes and identity management.
8. **Software and Data Integrity Failures** – Testing resilience against unauthorized data changes.
9. **Insufficient Logging and Monitoring** – Examining logging capabilities for effective threat detection.
10. **Server-Side Request Forgery (SSRF)** – Analyzing the server's handling of outbound requests.

In addition to OWASP guidelines, the testing also assessed encryption security, application configuration, user input validation, and potential server-side risks, ensuring no aspect of the system was overlooked.

For more information about the OWASP Top 10, visit their official website.

This comprehensive and methodical approach provided an invaluable understanding of our application's security posture, enabling us to proactively mitigate any identified risks.

When the Test Was Conducted

The Web Application Penetration Test, conducted in **November-December 2025**, evaluated the latest version of our application and platform. This proactive approach allowed us to identify and address vulnerabilities ahead of key development milestones or deployments, underscoring our commitment to a secure and robust system for our users.

Vulnerability Classification

Vulnerabilities detected during testing are classified into one of three categories, as follows:

High Severity

Definition:
Vulnerabilities that pose an immediate and critical risk to the application's security, data, or infrastructure. Exploitation could lead to severe consequences, such as unauthorized access, data theft, or complete system compromise. These issues typically require urgent remediation.

Examples:

- SQL Injection allowing unauthorized access to sensitive data.
- Authentication bypass enabling attackers to impersonate users.
- Remote Code Execution (RCE) vulnerabilities.
- Unrestricted file upload leading to server compromise.
- Exposed sensitive data (e.g., passwords, API keys) in plaintext.

Impact:

- Loss of sensitive data.
- Significant financial or reputational damage.
- Compromise of application or underlying systems.

Response:
Immediate attention is required, and patches or mitigation must be implemented as a top priority.

Medium Severity

Definition:
Vulnerabilities that present a moderate risk to the application but may require specific conditions or advanced techniques to exploit. While they are less likely to cause immediate critical impact, they could still lead to significant issues if combined with other vulnerabilities.

Examples:

- Cross-Site Scripting (XSS) allowing attackers to inject malicious scripts.
- Security misconfigurations, such as verbose error messages revealing system details.
- Weak or easily guessable password policies.
- Outdated libraries or components with known vulnerabilities (non-critical).

Impact:

- Potential for data leakage or compromise.
- Escalation to high-risk attacks when combined with other vulnerabilities.

Response:
Address these issues promptly, prioritizing based on the likelihood of exploitation and the sensitivity of affected systems.

Low Severity

Definition:
Minor vulnerabilities or security concerns that pose limited risk and are typically difficult to exploit. These issues are often related to best practices or configurations that could improve security but are unlikely to lead to a direct compromise.

Examples:

- Missing HTTP security headers (e.g., X-Content-Type-Options, Content-Security-Policy).
- Information leakage through public resources (e.g., server version disclosure).
- Outdated software versions without known critical vulnerabilities.
- Weak encryption protocols on non-sensitive endpoints.

Impact:

- Minimal or negligible immediate risk.
- May contribute to a broader attack surface in the future, if not addressed.

Response:
Address these issues as part of regular maintenance and ongoing security improvements.

Summary of Test Results

No identified findings were assessed as having a material impact on customer security or experience.

We remain committed to continuously improving our security measures to ensure the highest standards of protection and reliability.

Independent Assessment Statement (Pentest Provider):

“The web application provides a generally solid security posture, with the majority of issues identified being of low or informational severity and relating primarily to configuration hardening. No critical or high severity vulnerabilities were identified during this assessment.”